Amendments to Claims:

Listing of Claims:

1. (currently amended): A method in a data processing system, wherein the data processing system comprises source code and the source code comprises a plurality of lines, the method comprising the steps of:

displaying simultaneously a graphical and a textual representation of the source code, wherein the graphical and the textual representations of the source code is generated from a language-neutral representation of the source code that includes a data structure having a source code interface (SCI) model, an SCI package, an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transient meta model, and

wherein the graphical representation has portions that correspond to the lines; initiating an automated process that processes each of the lines; and while the automated process processes each of the lines,

displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated.

2. (original): The method of claim 1, further comprising the step of compiling the line before displaying the portion of the graphical representation that corresponds to the line.

- 3. (original): The method of claim 1, wherein while the automated process processes each of the lines, the method further comprises the step of displaying the line of source code in a visually distinctive manner.
- 4. (original): The method of claim 1, wherein the graphical representation comprises a class diagram.
- 5 (original): The method of claim 1, wherein the graphical representation comprises a sequence diagram.
- 6. (currently amended): A method in a data processing system, wherein the data processing system comprises source code and the source code comprises a plurality of lines, the method comprising the steps of:

displaying simultaneously a graphical and a textual representation of the source

code,

wherein the graphical and the textual representations of the source code generated from a language-neutral representation of the source code that includes a data structure having a source code interface (SCI)

model, an SCI package, an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transient meta model, and

wherein the graphical representation has portions that correspond to the

lines; and

for each of the lines, displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it appears that progression through the code is animated.

- 7. (original): The method of claim 6, further comprising the step of compiling the line before displaying the portion of the graphical representation that corresponds to the line.
- 8. (original): The method of claim 6, wherein for each of the lines, the method further comprises the step of displaying the line of source code in a visually distinctive manner.
- 9. (original): The method of claim 6, wherein the graphical representation comprises a class diagram.
- 10. (original): The method of claim 6, wherein the graphical representation comprises a sequence diagram.
- processing system comprises source code and the source code comprises a plurality of lines, wherein a graphical and a textual representation of the source code are displayed simultaneously and generated from a language-neutral representation of the source code that includes a data structure having a source code interface (SCI) model, an SCI package, an SCI

class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transient meta model, the method comprising the steps of:

displaying a graphical representation of the plurality of lines such that at least one of the

lines is not represented in the graphical representation;
initiating an automated process on each of the lines of the source code;
receiving an indication to suspend the automated process when the automated process
encounters one of the lines that is represented in the graphical representation; and
while the automated process is being performed on each of the lines of source code,
determining

whether the line is represented in the graphical representation; and when it is determined that the line is represented in the graphical representation, suspending the

automated process.

- performed on each of the lines of source code, the method further comprises the step of displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is arimated.
- 13. (original): The method of claim 12, wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of

displaying the line of source code in a visually distinctive manner.

MACCORD MASON

- 14. (original): The method of claim 11, further comprising the step of compiling the line before determining whether the line is represented in the graphical representation.
- 15. (original): The method of claim 11, wherein the graphical representation comprises a class diagram.
- 16. (original): The method of claim 11, wherein the graphical representation comprises a sequence diagram.
- 17. (currently amended): A method in a data processing system, wherein the data processing system comprises source code and the source code comprises a plurality of lines, the method comprising the steps of:

wherein the graphical and the textual representation of the source code are generated from a language-neutral representation of the source code that includes a data structure having a source code interface (SCI) model, an SCI package an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transcer meta model, and

initiating an automated process to be performed on each of the lines of the source code; receiving an indication to suspend the automated process when the automated

process encounters a selected one of the lines;
while the automated process is being performed on each of the lines of source code,
determining

whether the line is the selected line; and when it is determined that the line is the selected

line, suspending the automated process.

- performed on each of the lines of source code, the method further comprises the step of displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated.
- 19. (original): The method of claim 18, wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of displaying the line of source code in a visually distinctive manner.
- 20. (original): The method of claim 17, further comprising the step of compiling the line before determining whether the line is the selected line.
- 21. (original): The method of claim 17, wherein the graphical representation comprises a class diagram.

- 22. (original): The method of claim 17, wherein the graphical representation comprises a sequence diagram.
- 23. (currently amended): A method in a data processing system, wherein the data processing system comprises source code and the source code comprises a plurality of lines, the method comprising the steps of

wherein the graphical and the textual representations of the source code are generated from a language-neutral representation of the source code that includes a datastructure having a source code interface (SCI) model, an SCI package an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transient meta model;

receiving an indication of a first of the plurality of lines of the source code; selecting a second of the plurality of lines of the source code;

determining whether the second line is the same as the first line; and when it is determined that the second line is not the same as the first line, displaying the graphical representation of the second line in a visually distinctive

manne

24. (original): The method of claim 23, wherein when it is determined that the second line is not the same as the first line, the method further comprises the step of displaying the

second line of the source code in a visually distinctive manner.

manner

25. (original): The method of claim 23, wherein when it is determined that the second line is not the same as the first line the method further comprises the steps of:

selecting a third of the plurality offlines of the source code;

determining whether the third line is the same as the first line; and

when it is determined that the third line is not the same as the first line,

displaying the graphical representation of the third line in a visually distinctive

26. (original): The method of claim 25, wherein when it is determined that the third line is not the same as the first line; the method further comprises the step of displaying the third line of the source code in a visually dismettive manner.

- 27. (original): The method of claim 23, wherein the graphical representation comprises a class diagram.
- 28. (original): The method official 23, wherein the graphical representation comprises a sequence diagram.
- 29. (currently amended): As methed in a data processing system, wherein the data processing system comprises source code and the source code comprises a plurality of lines, wherein a graphical and a textual representation of the source code are displayed

simultaneously and generated from a language neutral representation of the source code that includes a data structure having a source code interface (SCI) model, an SCI package, an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transient meta model, the method comprising the steps of:

displaying a graphical representation of the plurality of lines such that at least one of the

lines is not represented in the graphical representation;
initiating an automated process on each of the lines of the source code;
while the automated process is being performed on each of the lines of source code,
compiling the line;
determining whether the compiled line produces an error; and
when it is determined that the compiled line produces the error,

30. (original): The method of claim 29 wherein while the automated process is being performed on each of the lines of source code the method further comprises the step of displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is ar imated.

uspending the automated process.

31. (original): The method of claim 10 wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of

displaying the line of source code in a visually distinctive manner.

- 32. (original): The method of claim 20, wherein the graphical representation comprises a class diagram.
- 33. (original): The method of claim 29, wherein the graphical representation comprises a sequence diagram.
- 34. (currently amended): A method in a data processing system, wherein the data processing system comprises source code and the source code comprises a plurality of lines, the method comprising the steps of:

wherein the graphical and textual representation of the source code are generated from a language-neutral representation of the source code that includes a data structure having a source code interface (SCI) model, an SCI package, an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transferit meta model;

selecting one of the plurality of lines of the source code;

compiling the selected line;

determining whether the compiled line produces an error, and

when it is determined that the sounded line does not produce an error, displaying the

graphical representation of the selected line in a visually distinctive manner.

35. (original): The method of claim 34 wherein when it is determined that the compiled line does not produce an error, the method further comprises the step of displaying the selected line of source code in a visually distinctive manner.

36. (original): The method of claim is wherein when it is determined that the compiled line does not produce an error, the method further comprises the steps of: selecting a second of the plurality of lines of the source code; compiling the second line; determining whether the compiled second line produces an error; and when it is determined that the compiled second line does not produce an error, displaying the graphical representation of the second line in a visually

- 37. (original): The method of claim 55, wherein when it is determined that the compiled second line does not produce an error, the method further comprises the step of displaying the second line of source cade in a visually distinctive manner.
- 38. (original): The method of claim 55, wherein the graphical representation comprises a class diagram.

distinctive manne

- 39. (original): The method of claim 34, wherein the graphical representation comprises a sequence diagram.
- 40. (currently amended): A competer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system comprises source code and the source code comprises a plurality of lines, the method comprising the steps of

wherein the graphical anedextrial representation of the source code are enerated from a language-neutral representation of the source code that includes a data structure having a source code interface (SCI) model, an SCI package, an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transient mera model, and

wherein the graphical represents form as portions that correspond to the lines; initiating an automated process that processes each of the lines; and while the automated process processes each of the lines, displaying the portion of the graphical

representation that corresponds is the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated.

41. (original): The computer reactable medium of claim 40, wherein the method further comprises the step of compline the period before displaying the portion of the graphical

representation that corresponds to the line

- 42. (original): The computer-readable medium of claim 40, wherein while the automated process processes each of the lines, the method further comprises the step of displaying the line of source code in a visually distinctive manner.
- 43. (original): The computer-reachile medium of claim 40, wherein the graphical representation comprises a class diagram.
- 44. (original): The computer readable medium of claim 40, wherein the graphical representation comprises a sequence diagram.
- 45. (currently amended): Accomplifer readable medium containing instructions for controlling a data processing system to perform a method, the data processing system comprises source code and the source code comprises a plurality of lines, the method comprising the steps of

displaying simultaneously a graphical and a textual representation of the source code, wherein the graphical and textual representations of the source code are generated from a language-neutral representation of the source code that includes a data structure having a source code interface (SCI) model, an SCI package an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transient ratio model, and

wherein the graphical representation has portions that correspond to the lines; and for each of the lines, displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it appears that progression through the core is animated.

- 46. (original): The computer readable medium of claim 45, wherein the method further comprises the sterrest compiling the line before displaying the portion of the graphical representation the compessions to the line.
- 47. (original): The computer readable medium of claim 45, wherein for each of the lines, the method further computes the step of displaying the line of source code in a visually distinctive manner.
- 48. (original): The computer readable medium of claim 45, wherein the graphical representation comprises a class diagram.
- 49. (original): The composer readable medium of claim 45, wherein the graphical representation comprises a sequence diagram.
- 50. (currently amended): Computer-readable medium containing instructions for controlling a data processing, yes ein to perform a method, the data processing system comprises source code and the source code comprises a plurality of lines, wherein a graphical and a textual processing of the source code are displayed

simultaneously and generated from a language-neutral representation of the source code that includes a data structure having a source code interface (SCI) model, an SCI package, an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transient meta model, the method comprising the steps at

displaying a graphical representation of the plurality of lines such that at least one

of the lines is not represented in the graphical representation; initiating an automated process on each of the lines of the source code; receiving an indication to suspend the automated process when the automated process encounters one of the lines that is represented in the graphical representation; and

while the automated process is being performed on each of the lines of source code, determining whether the line is represented in the graphical representation; and when it is determined that the line is represented in the graphical representation, suspending the automated process.

51. (original): The computer real and imedium of claim 50, wherein while the automated process is being performed or each of the lines of source code, the method further comprises the step of displaying the policient at the graphical representation that corresponds to the line in a visually distinctive manner security appears that progression of the automated process is animated.

- 52. (original): The computer-realizable medium of claim 51, wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of displaying the line of source code in a visually distinctive manner.
- 53. (original): The computer-readiline medium of claim 50, wherein the method further comprises the step of compiling the line before determining whether the line is represented in the graphical representation.
- 54. (original): The computer reactible medium of claim 50, wherein the graphical representation comprises a class diagram.
- 55. (original): The computer readable medium of claim 50, wherein the graphical representation comprises a sequence diagram.
- 56. (currently amended): A computer readable medium containing instructions for controlling a data processing system to perform a method, the data processing system comprises source code and the source code comprises a plurality of lines, the method comprising the steps of:

displaying simultaneously a graphical and a textual representation of the source code; initiating an automated process to be performed on each of the lines of the source code; wherein the graphical and textual representations of the source code are

generated from a large age he tral representation of the source code that includes a data structure laying a source code interface (SCI) model. an SCI

package an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transient metamodel;

initiating an automated process to be performed on each of the lines of the source code;

receiving an indication to suspend the automated process when the automated process encounters a selected one of the lines; and

while the automated process is being performed on each of the lines of source code,

determining whether the line is the selected line; and

when it is determined that the line is the selected line, suspending the automated

process.

- 57. (original): The computer-reladable medium of claim 56, wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated.
- 58. (original): The computer-madable medium of claim 57, wherein while the automated process is being perint reason each of the lines of source code, the method further comprises the step of disclaying the line of source code in a visually distinctive manner.

- 59. (original): The computer readable medium of claim 56, wherein the method further comprises the step of compiling the line before determining whether the line is the selected line.
- 60. (original): The computer readable medium of claim 56, wherein the graphical representation computers class diagram.
- 61. (original): The computer-readable medium of claim 56, wherein the graphical representation comprises a sequence diagram.
- 62. (currently amended). A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system comprises source code and the source code comprises a plurality of lines, the method comprising the steps of

displaying simultaneously a graphical and a textual representation of the source code;

wherein the graphical and invital representations of the source code generated inom a larguage-neutral representation of the source code that includes a data invicture having a source code interface (SCI)

model, an SCI package, an SCI class and an SCI member, wherein the language-neutral representation of the source code is storedin a non-repository ranseril meta model;

receiving an indication of a first of the plurality of lines of the source code; selecting a second of the plurality of lines of the source code; determining whether the second line is the same as the first line; and

when it is determined that the second line is not the same as the first line,

displaying the graphical representation of the second line in a visually distinctive

manner

- 63. (original): The computer-readable medium of claim 62, wherein when it is determined that the second line is not the same as the first line, the method further comprises the step of displaying the second line of the source code in a visually distinctive manner.
- 64. (original): The correputer-readable medium of claim 62, wherein when it is determined that the second line is not the same as the first line, the method further comprises the steps of:

selecting a third of the fillurality of lines of the source code;

determining whether the third line is the same as the first line; and

when it is determined that the third line is not the same as the first line, displaying

the graphical representation of the third line in a visually distinctive manner.

65. (original): The computer-readable medium of claim 64, wherein when it is determined that the third line is nor the same as the first line, the method further

comprises the step of displaying the third line of the source code in a visually distinctive manner.

- 66. (original): The computer readable medium of claim 62, wherein the graphical representation comprises a class diagram.
- 67. (original): The computer readable medium of claim 62, wherein the graphical representation computer a sequence diagram.
- for controlling a data processing sisters to perform a method, the data processing system comprises source code and the source code comprises a plurality of lines, wherein a graphical and a textual emesentation of the source code are displayed simultaneously and generated from a sanguage-neutral representation of the source code that includes a data structure traving a source code interface (SCI) model, an SCI package, an SCI class and ap 301 green the language-neutral representation of the source code interface (SCI) model, an SCI package, an SCI class and ap 301 green the language-neutral representation of the source code in a non-repository transient meta model, the method comprising the stages is

displaying a graphical representation of the plurality of lines such that at least one

of the lines is not expresented in the graphical representation; initiating an automated process can acree of the lines of the source code; while the automated process is be regreerformed on each of the lines of source code,

determining whether the compiled line produces an error; and when it is determined that the compiled line produces the error, suspending the automated process.

- 69. (original): The computer-readable medium of claim 68, wherein while the automated process is being performed on each of the lines of source code, the method further comprises the step of displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated.
- 70. (original): The computer-readable medium of claim 69, wherein while the automated process is being perturned on each of the lines of source code, the method further comprises the step of cisplaying the line of source code in a visually distinctive manner.
- 71. (original): The computer-readable medium of claim 68, wherein the graphical representation comprises at lass diagram.
- 72. (original): The complete stadable medium of claim 68, wherein the graphical representation comprises a sequence diagram.

73. (currently amended). A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system comprises source code are the source code comprises a plurality of lines, the method comprising the steps of

displaying simultaneous it a graphical and a textual representation of the source code;

wherein the graphical and rexital representations of the source code are generated from a single ge-neutral representation of the source code that includes a data structure having a source code interface (SCI) model, an SCI parkage an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transfer master master master model;

selecting one of the plurality of lines of the source code; compiling the selected fires

determining whether the compiled line produces an error; and when it is determined that the compiled line does not produce an error, displaying

the graphical representation of the selected line in a visually distinctive manner.

74. (original): The computer real able medium of claim 73, wherein when it is determined that the compiled line does not produce an error, the method further comprises the step of displaying the selected line of senice sode in a visually distinctive manner.

75. (original): The computer-read ble medium of claim 73, wherein when it is determined that the compiled line does ned produce an error, the method further comprises the steps of:

selecting a second of the plimality of lines of the source code; compiling the second line;

determining whether the compiled second line produces an error; and when it is determined that the corrected second line does not produce an error, displaying the graphical representation of the second line in a visually distinctive

manner

- 76. (original): The computer read intermedium of claim 75, wherein when it is determined that the compiled second line coer not produce an error, the method further comprises the step of displaying the second line of source code in a visually distinctive manner.
- 77. (original): The computer reactive medium of claim 73, wherein the graphical representation comprises a class diagram.
- 78. (original): the computer reacable medium of claim 73, wherein the graphical representation comprises a sequence discretization.

79. (currently amended). A data processing system comprising:

a secondary storage device further comprising source code wherein the source code comprises a plurality of lines.

a memory device further comprising a program

that displays simultaneously a grantical and a textual representation of the plurality of lines,

wherein the graphical and extual representations of the source code are generated

data structure having a source code interface (SCI) model, an SCI mackage, am SCI class and an SCI member, wherein the languagemeutral representation of the source code is stored in a non-repository

such that at least one of the lines is not represented in the graphical representation,

that initiates are automated process in each of the lines of the source code,
that receives are indication to suspend the automated process when the automated
process

encouriers one of the lines that is represented in the graphical representation, that determines whether the line is represented in the graphical representation while the

automated process is being performed on each of the lines of source code, and that suspends life automated process when it is determined that the line is

represented in the graphical tepresentation, and a processor for running the program.

- 80. (original): The data processing system of claim 79, wherein while the automated process is being performed on each of the lines of source code, the program further displays the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated.
- 81. (original): The data processing extent of claim 80, wherein while the automated process is being performed on each of the trees of source code, the program further displays the line of source code in a visually distinctive manner.
- 82. (original): The data processing system of claim 79, wherein the program further compiles the line before determining whether the line is represented in the graphical representation.
- 83. (original): The data processing system of claim 79, wherein the graphical representation comprises a class diagram.
- 84. (original): The data processing existem of claim 79, wherein the graphical representation comprises a sequence diagram
 - 85. (currently amended): A data processing system comprising:

a secondary storage device tirrher comprising source code wherein the source code comprises a plurality of lines.

a memory device further complising a program

that displays simultaneously argraphical and a textual representation of the source

code,

gene ated ire is a language-neutral representation of the source code that inc ares a data structure having a source code interace (SC) model, an SCI package, an SCI class and an SCI member, wherein the language-neutral representation of the source code interace (SC) model, an anon-repository transient meta model,

that initiates an autemated process to be performed on each of the lines of the source code.

that receives an incication its suspend the automated process when the
automated ricess are counters a selected one of the lines, and
that determines whether the line is the selected line while the automated
process is be six performed on each of the lines of source code
and that suspends it's automated process when it is determined that the
line is

the leasted me; and

a processor for running the pure and

- 86. (original): The data processing system of claim 85, wherein while the automated process is being performed on each of the lines of source code, the program further displays the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that procession of the automated process is animated.
- 87. (original): The data processing a stem of claim 86, wherein while the automated process is being performed on each of the lines of source code, the program further displays the line of source code in a visually distinctive manner.
- 88. (original): The data processing system of claim 85, wherein the program further compiles the line before determining wherest the line is the selected line.
- 89. (original): The data processing a stem of claim 85, wherein the graphical representation comprises a class diagrams
- 90. (original): The data processing system of claim 85, wherein the graphical representation comprises a sequence liamen.
 - 91. (currently amended): A data theresising system comprising:
 - a secondary storage device further comprising source code wherein the source code comprises a plurality of lines.
 - a memory device further charge a program

that displays simultaneously a graphical and a textual representation of the source code,

wherein the graphical and textual representations of the source code

are generated from a language-neutral representation of the source that

includes a datastructure having a source code interface (SCI) model, an

SCI package an SCI class and an SCI member, wherein the language
neutral representation of the source code is stored in a non-repository

ransient metal mode.

that selects a second of the plurality of lines of the source code,
that determines whether the second line is the same as the first line, and
that displays the graphical representation of the second line in a visually
distinctive manner when it is determined that the second line is not the
same as the first lines and

a processor for running the pine grain

- 92. (original): The data processing system of claim 91, wherein when it is determined that the second line is not the same as the rest line, the program further displays the second line of the source code in a visually cisting the manner.
- 93. (original): The data processing system of claim 91, wherein when it is determined that the second line is not the same as the first line, the program further selects a third of the plurality of lines of the source cook, the first whether the third line is the same as the first

line, and when it is determined that the third line is not the same as the first line, the program displays the graphical representation of the third line in a visually distinctive manner.

94. (original): The data processing system of claim 93, wherein when it is determined that the third line is not the same as the first line, the program further displays the third line of the source code in a visually distinctive manner.

95. (original): The data processing system of claim 91, wherein the graphical representation comprises a class diagram.

96. (original): The data processing system of claim 91, wherein the graphical representation comprises a sequence diagram.

97. (currently amended): A data processing system comprising:

a secondary storage device further comprising source code wherein the source code comprises a plurality of lines;

a memory device further comprising a program

that displays simultaneously a graphical and a textual representation of the plurality of lines,

wherein the graphical and textual representations of the source code are generated from a language neutral representation of the source code that includes a data structure having a source code interface (Self model, an SCI package, an SCI class and an SCI

that initiates an automated process on each of the lines of the source

ċode,

that while the automated process is being performed on each of the

lines

of source code

the program compiles the line,

determines whether the compiled line produces an error, and when it is determined that the compiled line produces the error,

the

program suspends the automated process; and

a processor for running the program

98. (original): The data processing system of claim 97, wherein while the automated process is being performed on each of the lines of source code, the program further displays the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated.

99. (original): The data processing system of claim 98, wherein while the automated process is being performed on each of the lines of source code, the program further displays

the line of source code in a visually distinctive manner.

100. (original): The data processing system of claim 97, wherein the graphical representation comprises a class diagram.

101. (original): The data processing system of claim 97, wherein the graphical representation comprises a sequence diagram.

102. (currently amended): A data processing system comprising:

a secondary storage device further comprising source code wherein the source code comprises a plurality of lines.

a memory device further comprising a program

that displays simultaneously a graphical and a textual representation of the source code.

wherein the graphical and textual representations of the source code are generated from a language-neutral representation of the source code that includes a data structure having a source code interface (SCI) model, an SCI package, an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transient meta model,

that selects one of the plurality of lines of the source code,

that compiles the selected in

that determines whether the compiled line produces an error, and

that displays the graphical representation of the selected line in a visually distinctive manner when it is determined that the compiled line does not produce an error; and

a processor for running the program

- 103. (original): The data processing system of claim 102, wherein when it is determined that the compiled line does not produce an error, the program further displays the selected line of source code in a visually distinctive manner.
- determined that the compiled line does not produce an error, the program further selects a second of the plurality of lines of the source code, compiles the second line, determines whether the compiled second line produces an error, and when it is determined that the compiled second line does not produce an error, the program displays the graphical representation of the second line in a visually distinctive manner.
- 105. (original): The data processing system of claim 104, wherein when it is determined that the compiled second line does not produce an error, the program further displays the second line of source code in a visually distinctive manner.
- 106. (original): The data processing system of claim 102, wherein the graphical representation comprises a class diagrams.

107. (original): The data processing system of claim 102, wherein the graphical representation comprises a sequence diagram.

108. (currently amended): A system having source code wherein the source code comprises a plurality of lines, the system comprising:

means for displaying simultaneously a graphical and a textual representation of the source code,

wherein the graphical and textual representations of the source code generated from a language-neutral representation of the source code that includes a data structure having a source code interface (SCI)

model, an SCI package an SCI class and an SCI member, wherein the language-neutral representation of the source code is stored in a non-repository transient meta model, and

wherein the graphical representation has portions that correspond to the lines; means for initiating an automated process that processes each of the lines; and means for displaying the portion of the graphical representation that corresponds to the line in a visually distinctive manner such that it visually appears that progression of the automated process is animated while the automated process processes each of the lines.

109. (previously presented): The method according to claim 1, wherein the textual representation of the source code is obtained from the source code directly.

- 110. (previously presented): The method according to claim 6, wherein the textual representation of the source code is obtained from the source code directly.
- 111. (previously presented): The method according to claim 11, wherein the textual representation of the source code is obtained from the source code directly.
- 112. (previously presented): The method according to claim 17, wherein the textual representation of the source code is obtained from the source code directly.
- 113. (previously presented): The method according to claim 23, wherein the textual representation of the source code is obtained from the source code directly.
- 114. (previously presented): The method according to claim 29, wherein the textual representation of the source code is obtained from the source code directly.
- 115. (previously presented): The method according to claim 34, wherein the textual representation of the source code is obtained from the source directly.
- 116. (previously presented): The computer-readable medium according to claim 40, wherein the textual representation of the source code is obtained from the source directly.
- 117. (previously presented): The computer-readable medium according to claim 45, wherein the textual representation of the source code is obtained from the source directly.

- 118. (previously presented) The computer-readable medium according to claim 50, wherein the textual representation of the source code is obtained from the source directly.
- 119. (previously presented). The computer-readable medium according to claim 56, wherein the textual representation of the source code is obtained from the source directly.
- 120. (previously presented): The computer-readable medium according to claim 62, wherein the textual representation of the source code is obtained from the source directly.
- 121. (previously presented). The computer-readable medium according to claim 68, wherein the textual representation of the source code is obtained from the source directly.
- 122. (previously presented): The computer-readable medium according to claim 73, wherein the textual representation of the source code is obtained from the source directly.
- 123. (previously presented): The computer-readable medium according to claim 79, wherein the textual representation of the source code is obtained from the source directly.
- 124. (previously presented): The computer-readable medium according to claim 85, wherein the textual representation of the source code is obtained from the source directly.
- 125. (previously presented): The computer-readable medium according to claim 91, wherein the textual representation of the source code is obtained from the source directly.

126. (previously presented): The computer-readable medium according to claim 97, wherein the textual representation of the source code is obtained from the source directly.

127. (previously presented): The computer-readable medium according to claim 102, wherein the textual representation of the source code is obtained from the source directly.

128. (previously presented): The computer-readable medium according to claim 108, wherein the textual representation of the source code is obtained from the source directly.

This Page is Inserted by IFW Indexing and Scanning Operations and is not part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

□ BLACK BORDERS
□ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
□ FADED TEXT OR DRAWING
□ BLURRED OR ILLEGIBLE TEXT OR DRAWING
□ SKEWED/SLANTED IMAGES
□ COLOR OR BLACK AND WHITE PHOTOGRAPHS
□ GRAY SCALE DOCUMENTS
□ CRAY SCALE DOCUMENTS
□ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY

IMAGES ARE BEST AVAILABLE COPY.

☐ OTHER: ____

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.